AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (Currently Amended) In a multi-tier server system that includes a back end server at a first tier and ~~one or more~~a plurality of additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the ~~one or more~~ plurality of additional servers before the data objects can be used by the ~~one or more~~ plurality of middle tier servers, a method for deploying one or more data types from the back end server to the ~~one or more~~ plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising:

an act of creating a special table in a database of the back end server, the special table including one or more fields for storing data identifying data types used by the plurality of middle tier servers and corresponding code for enabling use of each of the data types, the database of the back end server acting as a repository for each data type used by any of the plurality of middle tier servers, and the back end server acting as a single and centralized source from which ~~all~~each of the plurality of middle tier servers obtains all data types used by any other of the plurality of middle tier servers and the corresponding code required to enable use of the data types by the ~~one or more~~plurality of middle tier servers;

an act of identifying a data type to be deployed from the back end server to ~~the~~one or more of the plurality of middle tier servers;

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including data obtained from the special table, including data identifying the data type, one or more definitions of the data type, and the code for enabling processing of data corresponding to the data type; and

an act of transmitting the extended assembly to the one or more middle tier servers of the plurality of middle tier servers in the multi-tier system such that the data type, as transmitted to and received by the one or more of the plurality of middle tier servers in the multi tier system, is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system.

2.   (Currently Amended) A method as recited in claim 1, further including an act of creating logic modules in ~~the~~ one or more of the plurality of middle tier servers that enable the one or more of the plurality of middle tier servers to query for the extended assembly.

3.   (Original) A method as recited in claim 1, wherein the back end server includes a relational database.

4.   (Previously Presented) A method as recited in claim 3, wherein the back end server comprises an SQL server.

5.   (Currently Amended) A method as recited in claim 1, wherein the one or more of the plurality of middle tier servers includes an email server.

6.   (Cancelled).

7.   (Currently Amended) A method as recited in claim 1, wherein the act of identifying the data type to be deployed includes determining that the one or more of the plurality of middle tier servers has requested the extended assembly, since the one or more of the plurality of middle tier servers are not yet enabled for the data type.

8.   (Currently Amended) A method as recited in claim 7, further including an act of adding a new middle tier server to the multi-tier system, and wherein the new middle tier server comprises the one or more of the plurality of middle tier servers that has requested the extended assembly.

9.   (Original) A method as recited in claim 1, further including an act of creating one or more object tables that are linked to the special table and that include additional information defining the data type to be deployed, such that the extended assembly also includes the additional information.

10.   (Currently Amended) In a multi-tier server system that includes a back end server at a first tier and ~~one or more~~a plurality of additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the ~~one or more~~ plurality of additional servers before the data objects can be used by the ~~one or more~~ plurality of middle tier servers, a method for deploying one or more data types from the back end server to the ~~one or more~~ plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising:

an act of modifying a special table in a database of the back end server, the special table including one or more fields for storing data that identifies data types used by the plurality of middle tier servers and includes corresponding code for enabling use of each of the data types, the database of the backend server being a repository for each data type used by any of the plurality of middle tier servers, and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers and the corresponding code required to enable use of the data types by the ~~one or more~~plurality of middle tier servers, the act of modifying including at least one of modifying the stored data within the one or more fields and adding new stored data to the one or more fields;

an act of identifying a data type to be deployed from the back end server to ~~the~~one or more of the plurality of middle tier servers;

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including at least one of the modified stored data and the new stored data as obtained from the special table, including data identifying the data type, and the executable code that, when executed, enables the one or more middle tier servers to process the modified stored data or the new stored data associated with the data type; and

an act of transmitting the extended assembly to the one or more middle tier servers of the plurality of middle tier servers in the multi-tier system such that the data type as transmitted to and received by the one or more of the plurality of middle tier servers in the multi tier system is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system.

11.   (Currently Amended) A method as recited in claim 10, further including an act of determining which of the ~~one or more~~plurality of middle tier servers should be sent the extended assembly.

12.   (Currently Amended) A method as recited in claim 11, wherein determining which of the ~~one or more~~plurality of middle tier servers should be sent the extended assembly comprises the acts of:

sending data associated with the data type to the one or more of the plurality of middle tier servers; and

receiving one or more requests for the extended assembly from the one or more of the plurality of middle tier servers upon the one or more of the plurality of middle tier servers identifying that the data associated with the data type cannot be processed at the one or more of the plurality of middle tier servers.

13.   (Original) A method as recited in claim 10, wherein the back end server includes a relational database.

14.   (Previously Presented) A method as recited in claim 10, wherein the back end server comprises an SQL server.

15.   (Currently Amended) A method as recited in claim 10, wherein the one or more of the plurality of middle tier servers includes an email server.

16.   (Original) A method as recited in claim 10, wherein the act of modifying includes adding new stored data corresponding to a new data type not previously enabled in the multi-tier system prior to adding the new stored data.

17.   (Currently Amended) In a multi-tier server system that includes a back end server at a first tier and ~~one or more~~ a plurality of additional servers at a middle ~~tier~~tier, each additional server using multiple types of data objects that must be defined on the ~~one or more~~ plurality of additional servers before the data objects can be used by the ~~one or more~~ plurality of middle tier servers, a method for deploying one or more data types from the back end server to the ~~one or more~~ plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising:

an act of adding a new middle tier server to the multi-tier system, the new middle tier server being configured to utilize extended assemblies that are obtained from the back end server, the back end server acting as a repository for storing each data type used by any of the plurality of middle tier servers, the back end server ~~which acts~~ acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers and corresponding code required to enable use of the data types by the ~~one or more~~ plurality of middle tier servers, the extended assemblies being configured to enable the use of one or more data types that are defined by data and enabled by executable code that is contained in the extended assemblies;

an act of determining which of the one or more data types are to be deployed from the back end server to the new middle tier server, wherein the act of determining is based at least in part on a request by the new middle tier server for data to enable use of one or more data types;

an act of obtaining one or more extended assemblies corresponding to the one or more data types that have been determined to be deployed, each of the one or more extended assemblies including data and executable code obtained from a special table stored in a database of the back end server, the special table including one or more fields for storing data identifying data types and corresponding code for processing data associated with the data types; and

an act of transmitting, to the new middle tier server, the one or more extended assemblies that correspond to the one or more data types that have been determined to be deployed, such that the one or more data types as transmitted to, and received by, the new middle tier server are consistent and compatible with one or more data types of the same kind

on other middle tier servers in the system, and which were received by the other middle tier servers from the back end server.

18.   (Previously Presented) A method as recited in claim 17, wherein the act of determining is further based at least in part on the new middle tier server identifying what other data types are supported, and identifying that the one or more data types to be deployed are not supported at the new middle tier server.

19.   (Cancelled).

20.   (Cancelled).

21.   (Currently Amended) In a multi-tier server system that includes a back end server at a first tier and ~~one or more~~ a plurality of additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the ~~one or more~~ plurality of additional servers before the data objects can be used by the ~~one or more~~ plurality of middle tier servers, a method for deploying one or more data types from the back end server to the ~~one or more~~ plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising:

an act of creating a special table in a database of the back end server, the special table including one or more fields for storing data identifying a data type used by the plurality of middle tier servers and corresponding executable code for enabling processing of data associated with the data type, the database of the back end server acting as a repository for storing each data type used by any of the plurality of middle tier servers, and the back end server acting as a single and centralized source from which all middle tier servers obtain all data types used by any other of the plurality of middle tier servers and the corresponding code required to enable use of the data types by the ~~one or more~~ plurality of middle tier servers;

a step for deploying the data type from the back end server to ~~the~~ one or more of the plurality of middle tier servers, upon request, such that the data type as transmitted to and received by the one or more of the plurality of middle tier servers in the multi-tier server system is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system.

22. (Currently Amended) A method as recited in claim 21, wherein the step for deploying the data type to the one or more middle tier servers upon request comprises corresponding acts that include:

an act of identifying the data type to be deployed based on receipt of the data type at the one or more of the plurality of middle-tier servers, and the one or more of the plurality of middle-tier servers requesting an extended assembly for the data type since the data type cannot be processed at the one or more of the plurality of middle tier servers;

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including the data from the special table identifying the data type and the executable code for enabling processing of the data associated with the data type; and

an act of transmitting the extended assembly to the one or more of the plurality of middle tier servers in the multi-tier system that requested the extended assembly.

23. (Currently Amended) A method as recited in claim 22, further including an act of creating logic in the one or more of the plurality of middle tier servers that enables utilization of the extended assembly.

24. (Currently Amended) A method as recited in claim 22, further including an act of creating at least one object table in the database of the back end server that includes at least some information defining the data type, and wherein the extended assembly includes the at least some information.

25.　　(Currently Amended) A computer program product for use in a multi-tier server system that includes a back end server at a first tier and ~~one or more~~ a plurality of additional servers at a middle tier, each additional server using multiple types of data objects that must be defined on the ~~one or more~~ plurality of additional servers before the data objects can be used by the ~~one or more~~ plurality of middle tier servers, the computer program product including one or more computer-readable storage media having stored thereon computer-executable instructions for implementing a method for deploying one or more data types from the back end server to the ~~one or more~~ plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type as stored on each middle tier server in the system, the method comprising:

an act of creating a special table in a database of the back end server, the special table including one or more fields for storing data identifying data types used by the plurality of middle tier servers and corresponding code for enabling use of each of the data types, the database of the back end server acting as a repository storing each data type used by any of the plurality of middle tier servers, and the back end server acting as a single and centralized source from which ~~all~~ each of the plurality of middle tier servers obtains all data types used by any other of the plurality of middle tier servers and the corresponding code required to enable use of the data types by the ~~one or more~~ plurality of middle tier servers;

an act of identifying a data type to be deployed from the back end server to one or more of the plurality of middle tier servers;

an act of obtaining an extended assembly that corresponds to the data type to be deployed, the extended assembly including data obtained from the special table, including data identifying the data type, one or more definitions of the data type, and the code for enabling processing of data associated with the data type; and

an act of transmitting the extended assembly to ~~the~~ one or more of the plurality of middle tier servers in the multi-tier system such that the data type as transmitted to and received by the one or more of the plurality of middle tier servers in the multi tier system is consistent and compatible with a data type of the same kind stored on other middle tier servers in the system.

26.   (Currently Amended) A computer program product as recited in claim 25, wherein the method further includes an act of creating logic modules in the one or more of the plurality of middle tier servers that enable the one or more of the plurality of middle tier servers to query for the extended assembly.

27.   (Previously Presented) A computer program product as recited in claim 25, wherein the back end server includes an SQL server.

28.   (Currently Amended) A computer program product as recited in claim 25, wherein the one or more of the plurality of middle tier servers includes an email server.

29.   (Cancelled).

30.   (Currently Amended) A computer program product as recited in claim 25, wherein the act of identifying the data type to be deployed includes determining that the one or more of the plurality of middle tier servers has requested the extended assembly, since the one or more of the plurality of middle tier servers are not yet enabled for the data type.

31.   (Currently Amended) A computer program product as recited in claim 25, wherein the method further includes an act of adding a new middle tier server to the multi-tier system, and wherein the new middle tier server comprises the one or more of the plurality of middle tier servers that has requested the extended assembly.

32.   (Original) A computer program product as recited in claim 25, wherein the method further includes an act of creating one or more object tables that are linked to the special table and that include additional information defining the data type to be deployed, and wherein the extended assembly also includes the additional information.

33.   (Previously Presented) A computer program product as recited in claim 32, wherein the method further includes modifying at least one of the special table and the one or more object tables.

34.   (Currently Amended)   A method as recited in claim 1, wherein the extended assembly is a single data structure that includes all the data required to enable the one or more <u>of the plurality of</u> middle tier servers to use the data type.

35.   (Currently Amended) A method as recited in claim 1, wherein the one or more <u>of the plurality of</u> middle tier servers have limited program code means to process data associated with less than all of the data types in the multi-tier system, and the back end server has all program code means to process any data associated with all of the data types in the multi-tier system.

36.   (Currently Amended) A method as recited in claim 35, wherein the one or more <u>of the plurality of</u> middle tier servers are only equipped to recognize and process data objects associated with a particular data type when program code means comprising executable machine code of the extended assembly for the particular data type has been received from the back end server and installed at the one or more <u>of the plurality of</u> middle tier servers.

37.   (Currently Amended) At a middle tier server in a multi-tier database server system that includes a back end database server at a first tier and ~~one or more~~ a plurality of additional database servers at a middle tier, wherein the plurality of middle tier servers are ~~is~~ configured to process data corresponding to data types defined by the back end server at the first tier, a method for deploying one or more data types from the back end server ~~at the~~ to the plurality of middle tier servers in a manner that maintains consistency and compatibility in the definitions of the data types and in code associated with each data type in the multi-tier database server system, the method comprising:

an act of receiving at a middle tier server one or more data objects from a back end server, the one or more received data objects associated with at least one data type;

an act of initiating one or more processing functions for the one or more received data objects associated with the at least one data type;

an act of identifying that the at least one data type of the one or more data objects is not recognized, such that the initiated one or more initiated processing functions have failed at the middle tier server;

an act of pulling one or more extended assemblies corresponding to the at least one data type from the back end server, wherein the one or more extended assemblies were generated by the back end server in response to a request from the middle tier server to pull the one or more extended assemblies, wherein the back end server acts as a repository for each data type used by any of the plurality of middle tier servers and as a single and centralized source from which each of the plurality of middle tier servers obtains all extendable assemblies corresponding to all data types used by any other of the plurality of middle tier servers; and

an act of processing the one or more data objects associated with the at least one data type using the pulled one or more extended assemblies, wherein the middle tier server successfully recognizes the at least one data type, and successfully processes the one or more received data objects associated with the at least one data type.

38.   (Previously Presented) A method as recited in claim 37, wherein the one or more pulled extended assemblies comprise computer-executable instructions that, when executed at the middle tier server, cause one or more processors at the middle tier server to format the one or more data objects so that the one or more data objects can be processed.